UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/034,219	12/28/2001	Toni Juan	1662-47000 (P01-3694)	9074

22879 7590 06/27/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/27/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/034,219	JUAN ET AL.	
	Examiner	Art Unit	
	Aimee J. Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 April 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,10-14,19-24 and 26-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,10-14,19-24 and 26-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-5, 10-14, 19-24, 26-27, and new claims 28-37 have been considered. Claims 1, 10, 19, and 24 have been amended as per Applicant's request. New claims 28-37 have been added as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE as received on 01 April 2005 and Amendment as received on 01 April 2005.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-5 and 28-37 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Each of the claims has a similar limitation "applying said poison value through the store set dependence to subsequent load/store order violation occurrences to avoid executing and then re-playing a subsequent instruction subjected to a load/store order violation (Claim 1)." This limitation means that instructions subsequent to the load that needed to be re-processed due to a poisoned value are not to be re-executed, i.e. replayed. However, the application's specification states on page 16, paragraph 58 (emphasis added)

> In step 208, LOAD2 issues after STORE1 due to the store set dependence that was created as described above. However, the store sets poison table 675 is used to poison LOAD2 by propagating the poison information through the store set dependence (step 210). In step 212, the LOAD1 instruction reissues as a result of a cache fill and clears the register R1 poison tag. This permits the STORE1 instruction to then reissue in step 214 due to the register dependence on R1. Finally, the LOAD2 instruction reissues (step 216) due to the store set dependence with respect to STORE1.

5. An instruction subsequent to the reprocessed load has been executed and re-played.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-5, 10-14, 19-24, and 26-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant et al., U.S. Patent No. 6,665,792 (herein referred to as Merchant) in view of Abramson et al., U.S. Patent Number 5,881,262 (herein referred to as Abramson).

8. Regarding claims 1, taking claim 1 as exemplary, Merchant has taught a method of processing instructions in a microprocessor, comprising:

a. Fetching instructions from instruction memory (Merchant Col.3 line 54 – Col.4 line 8), certain fetched instructions being load instructions (loads) and causing load operations, and other fetched instructions being store instructions (stores) and causing store operations (Merchant Col.1 line 56 – Col.2 line 2),

b. Executing the fetched instructions out of program order (Merchant Col.6 lines 4-16),

c. Detecting a load/store order violation wherein a load executes prior to a store on whose data the load depends (Merchant Col.7 lines 28-45 and Col.8 line 41 - Col.9 line 30),

d. Creating a store set comprising a store set identifier value that identifies the store of the load/store order violation, and wherein the store set identifier links the load to the store (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43),

e. Saving the store set to the store set identifier table (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43). Here, a "store set" corresponding to entries in the load buffer, the store buffer and the bus queue is created upon detecting of a load/store order violation by making entries for the corresponding load and store instructions in their respective buffers (Merchant Col.3 lines 25-40), as well as setting the invalid store flag in the store buffer (Merchant Col.11 lines 1-41) and the inhibit load flag in the bus queue (Merchant Col.12 lines 3-22), with the entries in the buffers being indexed by the sequence number and source/destination addresses of the instructions (Merchant Col.12 lines 26-43).

f. Determining whether the store is poisoned by a previously poisoned instruction (Merchant Col.11 lines 13-41). Here, it is determined if there have been more than one store instructions with invalid (poison) flags, which causes the prior load instructions to be replied (Merchant Col.11 lines 26-31). Thus, the current store instruction has been poisoned (invalidated) by a previously poisoned (invalidated) instruction.

g. If the store is poisoned, setting a poison value that indicates through a store set dependence that the store is poisoned (Merchant Col.11 lines 13-41). Here, the store invalid flag (Merchant 510 of Fig.5) acts as a poison value (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

h. Re-processing said load if said poison value associated with said store indicates through the store set dependence that the store has been poisoned (Merchant Col.11 lines 13-41); and

i. Applying said poison value through the store set dependence to subsequent load/store order violation occurrences (Merchant column 11, lines 13-41).

9. Merchant has not explicitly taught said store set prevents a load from executing before a corresponding store and avoiding executing and then re-playing a subsequent instruction subjected to a load/store order violation. However, Merchant has taught that proper ordering of memory operations must be ensured (Merchant column 5, lines 42-45). Abramson has taught preventing a load from executing before a corresponding store and avoiding executing and then re-playing a subsequent instruction subjected to a load/store order violation (Abramson column 2, lines 39-53; column 4, lines 30-48 and 55-55-67; and Figure 1). In regards to Abramson, when a load is prevented from executing prior to resolution of the corresponding store, no re-play of the instruction is required since it was not executed using incorrect data and there is no possibility of incorrect processing for that load instruction. A person of ordinary skill in the art at the time the invention was made, and as taught by Abramson, that executing a load prior to the store it is dependent on would produce incorrect results and improper execution (Abramson column 2, lines 48-53). Therefore, it would have been obvious to a person of ordinary skill in

the art at the time the invention was made to incorporate preventing a load from executing before a corresponding store, as taught by Abramson, in the device of Merchant to ensure proper execution.

10. Regarding claims 10 and 19, taking claim 10 as exemplary, Merchant has taught a computer system, comprising:

a. A microprocessor (Merchant 100 of Fig.1) comprising a store set identifier table (Merchant column 6, lines 4-16 and 40-49 and Figures 2 and 3),

b. An input device coupled to said microprocessor (Merchant IQ, 112 of Fig.1),

c. Memory coupled to said microprocessor, said memory containing executable instructions (Merchant 104 of Fig.1),

d. Wherein said microprocessor:

i. Fetches instructions from said memory (Merchant Col.3 line 54 – Col.4 line 8), certain fetched instructions being load instructions (loads) and causing load operations, and other fetched instructions being store instructions (stores) and causing store operations (Merchant Col.1 line 56 – Col.2 line 2),

ii. Executes the fetched instructions out of program order (Merchant Col.6 lines 4-16),

iii. Detects a load/store order violation wherein a load executes prior to a store on whose data the load depends (Merchant Col.7 lines 28-45 and Col.8 line 41 - Col.9 line 30),

iv. Creates a store set comprising a store set identifier value that identifies the store of the load/store order violation, and wherein the store set identifier links the load to the store (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43),

v. Saves the store set to the store set identifier table (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43). Here, a "store set" corresponding to entries in the load buffer, the store buffer and the bus queue is created upon detecting of a load/store order violation by making entries for the corresponding load and store instructions in their respective buffers (Merchant Col.3 lines 25-40), as well as setting the invalid store flag in the store buffer (Merchant Col.11 lines 1-41) and the inhibit load flag in the bus queue (Merchant Col.12 lines 3-22), with the entries in the buffers being indexed by the sequence number and source/destination addresses of the instructions (Merchant Col.12 lines 26-43).

vi. Determines whether the store is poisoned by a previously poisoned instruction (Merchant Col.11 lines 13-41). Here, it is determined if there have been more than one store instructions with invalid (poison) flags, which causes the prior load instructions to be replied (Merchant Col.11 lines 26-31). Thus, the current store instruction has been poisoned (invalidated) by a previously poisoned (invalidated) instruction.

vii. If the store is poisoned, sets a poison value that indicates through a store set dependence that the store is poisoned (Merchant Col.11 lines 13-41).

Here, the store invalid flag (Merchant 510 of Fig.5) acts as a poison value (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

viii. Re-processes said load if said poison value associated with said store indicates through the store set dependence that the store has been poisoned (Merchant Col.11 lines 13-41); and

ix. Applies said poison value through the store set dependence to subsequent load/store order violation occurrences (Merchant column 11, lines 13-41).

11. Merchant has not explicitly taught preventing a load from executing before a corresponding store. However, Merchant has taught that proper ordering of memory operations must be ensured (Merchant column 5, lines 42-45). Abramson has taught preventing a load from executing before a corresponding store (Abramson column 2, lines 39-53; column 4, lines 30-48 and 55-55-67; and Figure 1). A person of ordinary skill in the art at the time the invention was made, and as taught by Abramson, that executing a load prior to the store it is dependent on would produce incorrect results and improper execution (Abramson column 2, lines 48-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate preventing a load from executing before a corresponding store, as taught by Abramson, in the device of Merchant to ensure proper execution.

12. Claim 19 is nearly identical to claim 10. Claim 19 lacks the input device of claim 10. However, claim 19 encompasses the same scope as claim 10. Therefore, claim 19 is rejected for the same reasons as claim 10.

13. Regarding claims 2, 11 and 20, taking claim 11 as exemplary, Merchant has taught the system of claim 10, wherein said poison value comprises a bit in a table (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

14. Claims 2 and 20 are nearly identical to claim 11, differing in their parent claims, but encompassing the same scope as claim 11. Therefore, claims 2 and 20 are rejected for the same reasons as claim 11.

15. Regarding claims 3, 12 and 21, taking claim 12 as exemplary, Merchant has taught the system of claim 10, wherein the store set includes a pointer that points to the poison value (Merchant Col.12 lines 26-43). Here, since there is a pointer to a store instruction (destination address) that is contained in the store set tables, and the poison value is directly associated with the store instruction (Merchant Fig.5 and Col.10 lines 54-59), there is a pointer to the poison value associated with that store instruction.

16. Claims 3 and 21 are nearly identical to claim 12, differing in their parent claims, but encompassing the same scope as claim 12. Therefore, claims 3 and 21 are rejected for the same reasons as claim 12.

17. Regarding claims 4, 13 and 22, taking claim 13 as exemplary, Merchant has taught the system of claim 10, wherein said store set includes a pair of tables which are used to identify said store instruction (Merchant Col.11 lines 22-41 and Col.12 lines 26-43). Here, entries in the load buffer, the memory ordering buffer, and the load buffer are all used in identifying an invalid store instruction.

18. Claims 4 and 22 are nearly identical to claim 13, differing in their parent claims, but encompassing the same scope as claim 13. Therefore, claims 4 and 22 are rejected for the same reasons as claim 13.

19. Regarding claims 5, 14 and 23, taking claim 14 as exemplary, Merchant has taught the system of claim 13, wherein said microprocessor clears said poison value when said store is no longer poisoned (Merchant Col.3 lines 28-39).

20. Claims 5 and 23 are nearly identical to claim 14, differing in their parent claims, but encompassing the same scope as claim 14. Therefore, claims 5 and 23 are rejected for the same reasons as claim 14.

21. Regarding claim 24, Merchant has taught a computer system, comprising:

a. A fetch stage which fetches instructions including a store instruction (store) and a load instruction (load) that target a common memory location (Merchant Col.3 line 54 – Col.4 line 8),

b. A store set identifier table coupled to said fetch stage (Merchant column 6, lines 4-16 and 40-49 and Figures 2 and 3) that comprises a store set associated with said load, wherein said store set comprises a store set identifier that identifies said store of a load/store violation (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43); and

c. Logic coupled to said fetch stage and said store set identifier table (Merchant column 6, line 65 to column 7, line 2), said logic

i. Determines if the data to be written by said store is stale (Merchant Col.7 lines 28-45 and Col.8 line 41 - Col.9 line 30),

ii. If said data is stale, sets a value associated with said store (Merchant Col.11 lines 13-41) and re-processes said load to execute after said data is no longer stale (Merchant Col.11 lines 13-41). Here, the store invalid flag (Merchant 510 of Fig.5) acts as a poison value (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

iii. Establishes said store set for said load to include said store, and saves said value associated with said store in said store set identifier table (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43). Here, a "store set" corresponding to entries in the load buffer, the store buffer and the bus queue is created upon detecting of a load/store order violation by making entries for the corresponding load and store instructions in their respective buffers (Merchant Col.3 lines 25-40), as well as setting the invalid store flag in the store buffer (Merchant Col.11 lines 1-41) and the inhibit load flag in the bus queue (Merchant Col.12 lines 3-22), with the entries in the buffers being indexed by the sequence number and source/destination addresses of the instructions (Merchant Col.12 lines 26-43).

22. Merchant has not explicitly taught said store set is usable to prevent a load from executing before a corresponding store that targets a common address. However, Merchant has taught that proper ordering of memory operations must be ensured (Merchant column 5, lines 42-45). Abramson has taught preventing a load from executing before a corresponding store that targets a common address (Abramson column 2, lines 39-53; column 4, lines 30-48 and 55-55-

67; and Figure 1). A person of ordinary skill in the art at the time the invention was made, and as taught by Abramson, that executing a load prior to the store it is dependent on would produce incorrect results and improper execution (Abramson column 2, lines 48-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate preventing a load from executing before a corresponding store, as taught by Abramson, in the device of Merchant to ensure proper execution.

23. Regarding claim 26, Merchant has taught the system of claim 24, wherein said microprocessor uses said store set to access said value (Merchant Col.12 line 62 – Col.13 line 10). Here, the processor has to access the memory order buffer and store buffer and the load buffer, which are the tables used in the store set, in order to check for an entry in the store buffer that is both older than a load instruction and has its invalid (poison) flag set.

24. Regarding claim 27, Merchant has taught the system of claim 24, wherein said value comprises a poison bit (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

25. Regarding claims 28 and 33, taking claim 33 as exemplary, Merchant has taught a system, comprising:

a. A microprocessor (Merchant 100 of Fig.1) comprising a store set identifier table (Merchant column 6, lines 4-16 and 40-49 and Figures 2 and 3),

b. Wherein said microprocessor:

i. Detects a load/store order violation wherein a load executes prior to a store on whose data the load depends (Merchant Col.7 lines 28-45 and Col.8 line 41 - Col.9 line 30),

ii. Forms a store set based on said detected load/store order violation (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43);

iii. Creates a store set comprising a store set identifier value that identifies the store of the load/store order violation, and wherein the store set identifier links the load to the store (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43),

iv. Determines whether the store is poisoned by a previously poisoned instruction (Merchant Col.11 lines 13-41). Here, it is determined if there have been more than one store instructions with invalid (poison) flags, which causes the prior load instructions to be replied (Merchant Col.11 lines 26-31). Thus, the current store instruction has been poisoned (invalidated) by a previously poisoned (invalidated) instruction.

v. If the store is poisoned, setting a poison value that indicates through a store set dependence that the store is poisoned (Merchant Col.11 lines 13-41). Here, the store invalid flag (Merchant 510 of Fig.5) acts as a poison value (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

vi. Applies said poison value through the store set dependence to subsequent load/store order violation occurrences (Merchant column 11, lines 13-41).

26. Merchant has not explicitly taught said store set prevents a load from executing before a corresponding store and avoiding executing and then re-playing a subsequent instruction subjected to a load/store order violation. However, Merchant has taught that proper ordering of

memory operations must be ensured (Merchant column 5, lines 42-45). Abramson has taught preventing a load from executing before a corresponding store and avoiding executing and then re-playing a subsequent instruction subjected to a load/store order violation (Abramson column 2, lines 39-53; column 4, lines 30-48 and 55-55-67; and Figure 1). In regards to Abramson, when a load is prevented from executing prior to resolution of the corresponding store, no re-play of the instruction is required since it was not executed using incorrect data and there is no possibility of incorrect processing for that load instruction. A person of ordinary skill in the art at the time the invention was made, and as taught by Abramson, that executing a load prior to the store it is dependent on would produce incorrect results and improper execution (Abramson column 2, lines 48-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate preventing a load from executing before a corresponding store, as taught by Abramson, in the device of Merchant to ensure proper execution.

27. Claim 28 is nearly identical to claim 10. Claim 28 lacks the microprocessor of claim 33. However, claim 28 encompasses the same scope as claim 33. Therefore, claim 28 is rejected for the same reasons as claim 33.

28. Referring to claims 29 and 34, Merchant has taught wherein the microprocessor forms said store set by assigning a store set identifier to the load and corresponding store involved in the load/store order violation (Merchant Col.3 lines 25-40, Col.11 lines 1-41 and Col.12 lines 3-22, 26-43).

29. Referring to claims 30 and 35, Merchant has taught storing said store set identifier value in at least two locations in a store set table, a first location associated with the load and a second

location associated with the corresponding store (Merchant Col.11 lines 22-41 and Col.12 lines 26-43).

30. Referring to claims 31 and 36, Merchant has taught wherein the microprocessor also sets a bit in the store set table to indicate that the store set identifiers are valid (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

31. Referring to claims 32 and 37, Merchant has taught storing poison information in a poison table indexed by the store set identifier from the store set table (Merchant Col.10 lines 54-59 and Col.11 lines 22-41).

Response to Arguments

32. Applicant's arguments with respect to claims 1-5, 10-14, 19-24, and 26-37 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

33. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

34. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

35. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
22 June 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100